McGinn & Gibb, PLLC
A Professional Limited Liability Company
Patents, Trademarks, Copyrights, and Intellectual Property Law
8321 Old Courthouse Road, Suite 200
Vienna, Virginia 22182-3817
Telephone (703) 761-4100
Facsimile (703) 761-2375; (703) 761-2376

APPLICATION FOR UNITED STATES LETTERS PATENT

APPLICANT: CHEN, ET AL.

FOR: SYSTEM AND METHOD FOR MONITORING EVENTS AGAINST CONTINUAL RANGE QUERIES

DOCKET NO.: YOR920030165US1

SYSTEM AND METHOD FOR MONITORING EVENTS AGAINST CONTINUAL RANGE QUERIES

Cross-Reference to Related Applications

The present Application is related to the following co-pending application:

U.S. Patent Application No. 10/ ___,___, filed on ________, to Chen et al., entitled "Method and Structure for Monitoring Moving Objects", having IBM Docket YOR920030164US1; and

U.S. Patent Application No. 10/ ___,___, filed on ________, to Chen et al, entitled "System and Method for Indexing Queries, Rules and Subscriptions", having IBM Docket YOR920030265US1,

both assigned to the present assignee, and both incorporated herein by reference.

DESCRIPTION

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention generally relates to activity/event monitoring in various application areas such as business activity monitoring for corporate management, sensor activities monitoring for continual queries, road traffic condition monitoring for traffic control, event matching for pub/sub applications, information monitoring for selective information dissemination, and health activity monitoring for disease outbreaks or

bio-attacks. More specifically, it discloses a predicate/query indexing method for monitoring activities/events against a plurality of continual range predicates/queries.

Description of the Related Art

Fast matching of events against a large number of predicates/queries is important for many applications, such as business activity monitoring, content-based pub/sub (publication/subscription), continual queries, health activity monitoring, and selective information dissemination services. Users simply specify their interests in the form of a conjunction of predicates. The system then automatically monitors these user interests against a continual stream of events, conditions, or activities.

Generally, an efficient predicate index is needed. Prior work for fast event monitoring has mostly focused on building predicate indexes with equality-only clauses, as in, for example:

- "Matching events in a content-based subscription system," by M. K. Aguilera et al., in *Proc. of Symposium on Principles of Distributed Computing*, 1999; and
- "Filtering algorithms and implementation for very fast publish/subscribe systems," by F. Fabret et al., in *Proc. of ACM SIGMOD*, 2001.

However, many queries/predicates contain non-equality range clauses. For example, stock price, salary, and object location tend to involve non-equality range predicates.

It is difficult to construct an effective index for multidimensional range predicates. It is even more challenging if these predicates are overlapping, as they usually are because people tend to share similar interests. For instance, people tend to be interested in the

current price ranges of individual stocks. Hence, the range predicates of their interests are likely to be overlapping.

Although multidimensional range predicates can be treated as spatial objects, a typical spatial index, such as an R-tree, is generally not effective for monitoring events. This is because an R-tree method is generally a disk-based indexing method and an R-tree quickly degenerates if spatial objects are highly overlapping (V. Gaede et al., "Multidimensional access methods," *ACM Computing Surveys*, 30(2):170-231, 1998.; A. Guttman, "R-trees: A dynamic index structure for spatial searching," *Proceedings of ACM SIGMOD*, 1984.)

Hence, a need is recognized for a new and effective system and method for efficient monitoring of events against range queries, some of them may overlap with one another.

SUMMARY OF THE INVENTION

In view of the foregoing problems, drawbacks, and disadvantages of the conventional systems, it is an exemplary feature of the present invention to provide a structure (and method) for building an efficient query index for monitoring continual range queries against events.

It is, therefore, an exemplary purpose of the present invention to provide a structure and method for application areas such as business activity monitoring for corporate management, sensor activities monitoring for continual queries, road traffic condition monitoring for traffic control, event matching for publication/subscription applications, information monitoring for selective information dissemination, and health activity

monitoring for disease outbreaks or bio-attacks, using an efficient query index for monitoring continual range queries against events.

Hence, in a first aspect of the present invention, described herein is a method (and structure) for monitoring continual range queries against events includes decomposing each range query into one or more predefined virtual constructs, building a query index, and using the query index to match an event with the range queries.

In a second aspect of the present invention, described herein is a method of providing a service of monitoring events or conditions, including at least one of: providing a service that monitors events against interests of a customer, the service monitoring the events by decomposing continual range queries related to the customer interests into one or predefined virtual constructs, building a query index, and using the query index to match an event with said range queries; maintaining one or more customer interests expressed as continual range queries for a service that monitors events in the manner described; and notifying a subset of the customers whose interests match an event.

In a third aspect of the present invention, described herein is a system for monitoring continual range queries against events, including a decomposing module that decomposes each range query with one or more predefined virtual constructs, a query index construction module, and an event matching module that uses the query index to match an event with the range queries.

In a fourth aspect of the present invention, described herein is an apparatus for monitoring continual range queries against events, wherein the apparatus is one of: a query monitor that includes a decomposing module that decomposes each range query into one or

more predefined virtual constructs, a query index construction module, and an event matching module that uses the query index to match an event with the range queries; a sensor to detect occurrence of events and provides the occurrence of events into the query monitor; and a client receiver to permit a client to be notified of occurrence of an event of interest to the client.

According to a fifth aspect of the present invention, described herein is signal-bearing medium tangibly embodying a program of machine-readable instructions executable by a digital processing apparatus to perform the method for monitoring continual range queries against events, as described above.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, aspects and advantages will be better understood from the following detailed description of a preferred embodiment of the invention with reference to the drawings, in which:

Figure 1 shows an exemplary system block diagram 100 of an environment where a plurality of continual range queries are monitored against events;

Figure 2 shows a flowchart 200 that summarizes an exemplary embodiment of the present invention;

Figure 3 shows an example of a set 300 of nine Virtual Construct Rectangles (VCRs) whose bottom-left corners are positioned at point (0, 0);

Figure 4 shows an example of assigning a unique identification (ID) to any VCR in an area 400 being monitored;.

Figure 5 shows an example of decomposing a range query 500;

Figure 6 shows a flow chart 600 of an exemplary decomposition routine;

Figure 7 shows an example 700 of a covering VCR set; and

Figure 8 shows an exemplary flow chart 800 of an exemplary routine for finding all the IDs of VCRs in the covering VCR set of a point;

Figure 9 illustrates an exemplary hardware/information handling system 900 for incorporating the present invention therein; and

Figure 10 illustrates a signal bearing medium 1000 (e.g., storage medium) for storing steps of a program of a method according to the present invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT OF THE INVENTION

Referring now to the drawings, and more particularly to Figure 1, an exemplary embodiment will now be described. The present invention teaches a novel indexing scheme for range queries in an event space, exemplarily, a two-dimensional space. An example of a basic construct in the present invention is the "VCR", which stands for "Virtual Construct Rectangle". This basic construct can be extended to spaces having more than two dimensions. For the single dimensional space, a virtual construct interval (VCI) scheme can

be used, as further described in the second of the above-identified copending patent applications.

Figure 1 shows an exemplary system block diagram 100 of an environment where a plurality of continual range queries are monitored against events. One or more sensors 101, 102 are deployed to monitor conditions, events, or activities. These events or activities are sent to one or more query monitors 121, 122, via a communication network 100. Range queries are submitted from one or more clients 111, 112 to the query monitors 121, 122.

The sensors, clients and query monitors are connected via a communication network 100, e.g., the Internet. The query monitors 121, 122 typically are computer servers. The query indexing method disclosed in the present invention is employed by the query monitors 121, 122 to efficiently identify all the range queries that match an incoming event. Those skilled in the art will appreciate that the sensors, the clients, or the query monitors may employ wireless technologies for communication.

Figure 2 shows a flowchart 200 that summarizes an exemplary embodiment using the novel VCR indexing scheme for range queries in two-dimensional space. As mentioned previously, VCR stands for "Virtual Construct Rectangle". The concept discussed herein can readily be extended to more than two dimensional spaces.

As shown in Figure 2, in steps 201 and 202, an empty ID list is established and a set of virtual construct rectangles are predefined (e.g., a set of rectangles having sides based on powers of 2). In steps 203 and 204, upon entry, each range query is decomposed into one or more of the predefined VCRs. In step 205, the range query ID is then inserted into the ID lists associated with these decomposed VCRs.

As shown in steps 206-209, event matching is conceptually simple. For each event point, the search results are stored in the ID lists associated with the activated VCRs that cover that point. However, it is computationally nontrivial to identify the covering VCRs, and the present invention provides a very efficient method to accomplish this task.

A covering VCR set for each point is defined. These covering VCR sets share two common properties: constant size and identical gap pattern. Based on these properties, a procedure for efficient event monitoring with a constant time complexity is disclosed.

For exposition, the two-dimensional case will be discussed. A set of virtual construct rectangles is defined so that each point in the monitoring region, which is a rectangle for the two-dimensional case, will be covered. As will be demonstrated shortly, the set of VCRs for these points in the monitoring region will be identified as based on location of the VCR's bottom-left corner and the VCR size, but the VCRs in the monitoring region may have different sizes and shapes. Each VCR has a unique ID and this ID can be computed with a simple formula, given the location of its bottom-left corner and its width and height and the width of the overall region being monitored.

Before a range query, which is represented as a rectangle in the two-dimensional case, is inserted into the ID list, it is decomposed into one or more VCRs (e.g., step 204). Then, in step 205, the query ID is inserted into the ID lists associated with the decomposed VCRs. There are many ways that range query decomposition could be done.

One simple way discussed shortly is to cut a strip rectangle from the bottom of the query rectangle and progressively move upwards. The height of the strip rectangle is the maximum VCR height that will fit into the range query. For each strip rectangle, the largest VCR having the same height as the strip rectangle is used to cut the strip rectangle, and each strip rectangle is decomposed with VCRs of the same height.

As shown in step 207 of Figure 2, to search all the queries that are matched by an incoming event (e.g., step 206), all the covering VCRs for that event point are found, and

then, from these covering VCRs, in step 208, all the queries that have used at least one of these covering VCRs in their decomposition is found and reported in step 209. The covering VCR set for a point is defined as the set containing all the VCRs that can possibly cover this point.

The covering VCR sets for all the points share two common properties, enabling an efficient way to enumerate all the IDs of VCRs in a covering VCR set. A distance table which stores the ID differences between a VCR in the covering VCR set and a pivot point is pre-computed. As a result, all the VCRs in a covering VCR set can be enumerated by adding the differences to the ID of the pivot point.

Finally, as shown in step 210, to delete a range query, it is first decomposed into one or more VCRs, similar to query insertion. Then, in step 211, the query ID is removed from the ID lists associated with the decomposed VCRs.

The Virtual Construct Rectangles (VCRs) used in this exemplary preferred embodiment is shown in Figure 3 for two-dimensional space. Extension to a higher dimension, as well as to a single dimension is straightforward.

The query indexing method exemplarily disclosed herein predefines B virtual construct rectangles for each point (x, y). Figure 3 shows an example of a set 300 of nine VCRs (e.g., VCRs identified as VCR:0 through VCR:8). In this scheme, the number of VCRs possible (e.g., $B = 9$) is based on the size of the largest VCR (e.g., the 4 x 4 size VCR labeled VCR:8, in Figure 3).

It can be seen that, in Figure 3, the bottom-left corner of each VCR is positioned at point (0, 0). It is also seen that each VCR has its own unique shape and size, ranging from 1 x 1 through 4 x 4. The differences in VCR size is interrelated in powers of 2. It should be apparent that using powers of 2 is inherently supported by the binary representation used in a

computer word, and has the advantage that the VCR concept allows a construction of VCRs that can cover any rectangular range query, as will be shortly apparent. Each VCR also has its own identification (ID) as well (e.g., ranging from 0 to 8 in Figure 3), to be discussed next.

In accordance with the exemplary identification scheme, the VCR IDs shown in Figure 3 are quite simple. The identification scheme will also allow a unique identification to be assigned to a VCR as it is activated to be assigned any arbitrary location. The equation for the exemplary identification scheme is based on the location of the lower left corner of the VCR in the monitored region, the width and height dimensions of the VCR, the size of the entire region being monitored (i.e., the overall area being scanned for check for event points), and the number B of VCRs in the VCR set (e.g., B = 9 in Figure 3, since there are 9 possible VCRs in the 4 x 4 set). However, it should also be apparent that different numbers of VCRs with different shapes and sizes can be defined for a point without departing from the intent of the present invention.

Figure 4 shows an example of assigning a unique ID to any VCR 401 in an area 400 being monitored (e.g., VCR$(x,y,2^i,2^j)$, where (x,y) represents the location of the VCR bottom-left corner). The monitoring area 400 is defined by a rectangular region, where $0 \leq x \leq R_x - 1$, and $0 \leq y \leq R_y - 1$. As shown by the formula 402 in Figure 4, and since there are *B* predefined VCRs, a unique ID can then assigned to any VCR $(x,y,2^i,2^j)$, where (x, y) represents the bottom-left corner position of the VCR, 2^i is the VCR width, and 2^j is the VCR height: $ID(x,y,2^i,2^j) = B(x + yR_x) + j(k_x + 1) + i$, where $B = (k_x + 1)(k_y + 1)$, $k_x = \log(L_x)$, and $k_y = \log(L_y)$. In the exemplary scheme, the logarithms would conveniently be base 2.

Therefore, returning to Figure 3, this formula can be checked out for various of the VCRs shown in that figure, as follows:

For "VCR:0", since L_x and L_y are 4, then $k_x = 2$. Also, $(x,y) = (0,0)$, $i = 0$, $j = 0$, $B = 9$. Therefore, VCR $(0,0,2^0,2^0) = 9(0 + 0 * Rx) + 0(2 + 1) + 0 = 0$. Note that the width of the monitor area Rx is irrelevant when the VCR is located at (0,0), as in Figure 3.

For "VCR: 4", $k_x = 2$, $(x,y) = (0,0)$, $i = 1$, $j = 1$, $B = 9$, so that VCR $(0,0,2^1,2^1) = 9(0 + 0 * Rx) + 1(2 + 1) + 1 = 4$.

For "VCR: 8", $k_x = 2$, $(x,y) = (0,0)$, $i = 2$, $j = 2$, $B = 9$, so that VCR $(0,0,2^2,2^2) = 9(0 + 0 * Rx) + 2(2 + 1) + 2 = 8$.

The formula above is based on the fact that there are $(x + yR_x)$ points prior to (x, y) in a horizontal scan 403 of the integer points in the monitoring region 400, as shown in the lower portion of Figure 4. That is, horizontal scanning of the monitored region 400 would occur by holding y constant and varying x through the monitored region (e.g., $x = 0 \rightarrow Rx\text{-}1$).

Moreover, for all the *B* VCRs sharing the same bottom-left corners, their IDs are assigned according to Figure 3, where L_x and L_y are the maximum width and height, respectively, for a VCR predefined set.

In the exemplary embodiment, L_x and L_y are assumed to be numbers that are a power of 2. Namely, $k_x = \log(L_x)$, and $k_y = \log(L_y)$, or more accurately, $k_x = \log_2(L_x)$, and $k_y = \log_2(L_y)$, where k_x and k_y are integers. However, it should be apparent that the IDs can be assigned differently. As one alternative method, the scan could be performed vertically (e.g., holding x constant and varying y from 0 to Ry-1.

With predefined VCRs, the range queries are decomposed with one or more VCRs. Figure 5 shows an example of decomposing an exemplary range query (3, 3, 11, 6), meaning a range 500 having bottom-left corner at (3, 3), width 11 and height 6. In this exemplary method, first two 4x4 VCRs 501, 502 are activated. Activation occurs via the decomposition process. Then, a 2x4 VCR 503, a 1x4 VCR 504, two 4x2 VCRs 505, 506, one 2x2 VCR 507 and one 1x2 VCR 508 are activated. Hence, a total of eight VCRs are used to decompose this exemplary range query 500.

Incidently, using the formula shown in Figure 4 for VCR identification, the identification of various of these eight VCRs (using the VCRs from Figure 3, where $k_x = 2$) would be (since B = 9, Rx = 17):

VCR_{501}, ID $(3,3,2^2,2^2) = 9(3 + 3 * 17) + 2(2 + 1) + 2 = 494$;

VCR_{502}, ID $(7,3,2^2,2^2) = 9(7 + 3 * 17) + 2(2 + 1) + 2 = 530$;

VCR_{504}, ID $(13,3,2^0,2^2) = 9(13 + 3 * 17) + 2(2 + 1) + 0 = 582$;

VCR_{505}, ID $(3,7,2^2,2^1) = 9(3 + 7 * 17) + 1(2 + 1) + 2 = 1103$; and

VCR_{508}, ID $(13,7,2^0,2^1) = 9(13 + 7 * 17) + 1(2 + 1) + 0 = 1713$.

Figure 6 shows an exemplary flow chart 600 of an exemplary decomposition routine. As shown in steps 601 and 602, to decompose a range query (*a, b, w, h*), the routine first initializes the working rectangle to be the range query and the decomSet to be empty. Step 603 causes the routine to then loop until the height of the working rectangle is less than or equal to 0.

When that happens, in step 608, the decomSet is returned, where decomSet stores all the IDs of VCRs that are used to decompose (*a, b, w, h*). In step 604, if the height of the

working rectangle is greater than 0, a strip rectangle with a height of maxVCRh(H_w) is cut from the working rectangle. That is, the height of the strip rectangle is the maximum VCR height that is smaller than or equal to the height of the working rectangle. In step 606, for this strip rectangle, the largest VCR that can decompose the strip rectangle from the left and moving towards the right is found and added to the decomSet.

After each strip rectangle is decomposed, in step 607, another strip rectangle is cut from the working rectangle and similarly decomposed.

At the end, in step 608, the decomposed VCRs are contained in decomSet. It should be apparent that there are other ways to decompose a range query. For example, some of the decomposed VCRs may overlap one another. It should also be apparent that it would be possible to expand the set of VCRs to include a larger power of 2, should an entered range query be larger than initially expected.

After decomposition, the query ID is inserted into the ID lists associated with the decomposed VCR (e.g., step 205 in Figure 2). As a result, the ID list associated with a VCR contains all the IDs of range queries that have used this VCR in its decomposition. Events or conditions matched by this said VCR are also matched by these range queries. Thus, monitoring range queries against events becomes the task of finding all the covering VCRs for any point, since events are represented by points in the monitoring region.

According, assuming that $Cov(a, b)$ represents the covering VCR set for point (a, b). That is, $Cov(a, b)$ contains all the VCRs that actually cover point (a, b). Figure 7 shows an example of $Cov(a, b)$.

Remembering that the VCR set contains a maximum VCR having dimensions Lx, Ly, then *Cov*(*a, b*) will include the set of any activated VCRs whose bottom-left corners are in the shaded region southwest of (*a, b*) and whose upper-right corners are in the shaded region northeast of (*a, b*), as shown in Figure 7. For example, 701, 702, 703, and 704 are such VCRs in the covering set.

The covering VCR sets for all points in the monitoring region share two common properties: constant size and identical gap pattern. For ease of exposition, consider the region that is inside the monitoring region separated with a boundary strip region, namely, the region $L_x \leq a \leq R_x - Lx - 1$, and $L_y \leq b \leq R_y - L_y - 1$. For the boundary strip region, a similar method can be applied. It is noted that a and b are considered variables in this context.

The constant size property says that $|Cov(a,b)|=|Cov(c,d)|$, for any two different points (*a, b*) and (*c, d*). Namely, the sizes of the covering VCR set for individual points are all the same. This can be visually appreciated from Figure 7 by moving around all the points together. The sizes of the shaded regions remain the same.

The identical gap pattern property says that, if the IDs inside a covering VCR set are sorted, the ID differences between any two VCRs are constant among all the covering VCR sets of different points. Let $V_{i,(a,b)}$ denote the ID of a covering VCR for (*a, b*) and $V_{i+1,(a,b)} > V_{i,(a,b)}$. Then, $V_{i+1,(a,b)} - V_{i,(a,b)} = V_{i+1,(c,d)} - V_{i,(c,d)}$, for $1 \leq i \leq |Cov(a,b)|$ and any two points (*a, b*) and (*c, d*). This property can also be appreciated from Figure 7 by first grouping all the drawings together as a unit and then moving it around. When the center is moved from (*a, b*) to (*c, d*), the relative positions of all the covering VCRs stay the same.

With these two properties, a difference table *DT* can be pre-computed, which table stores the ID differences between all the covering VCRs and a pivot VCR. For a point (*a*, *b*), the pivot VCR is defined as $(a - L_x, b - L_y, 1, 1)$.., and shown in Figure 7 as VCR_{705}. This pivot VCR is not a member of the covering VCR set for (*a*, *b*), however. Together with the pivot VCR, this DT can be used to efficiently compute all the IDs of VCRs in the covering VCR set of any point.

Figure 8 shows an exemplary flow chart 800 of an exemplary routine for finding all the IDs of VCRs in the covering VCR set of a point (*a*, *b*). In step 801, the routine first initializes *Cov*(*a*, *b*) to be empty, computes the ID *c* of the pivot VCR, and sets $i = 1$.

Then, in step 804, for each value in difference table *DT*, each value is added with the ID of the pivot VCR and included in *Cov*(*a*, *b*). At the end, in step 803, the covering VCR set stored in *Cov*(*a*, *b*) is returned.

With the IDs of all the covering VCRs for a point, the range queries can be found directly from the ID lists associated with all the covering VCRs.

Those skilled in the art will appreciate that the VCR indexing method can be extended to K > 2 dimensions. Assuming that the *K* dimensions have $R_1, R_2, ..., R_K$ values (all starting from 0), and also assuming that $L_1, L_2, ..., L_K$ are the maximum sizes of a *K*-dimensional virtual construct regions (VCR), *B* VCRs can potentially be defined for each point. Each VCR is assigned with a unique ID.

To insert a range query, it is first decomposed into a set of VCRs and then the query ID is inserted into the ID lists associated with the decomposed VCRs. To find all the range

queries matching an event, all the covering VCRs are first found and then the query IDs associated with the covering VCRs are found.

To delete a range query, it is first decomposed into one or more VCRs, similar to query insertion. Then, the query ID is removed from the ID lists associated with the decomposed VCRs.

Those skilled in the art will also appreciate that, in a single dimensional space, the virtual construct rectangles become virtual construct intervals (VCI), as further described in the second above-listed copending patent application.

Those skilled in the art will appreciate that various kinds of services can be provided based on the system and method disclosed in the current invention. For example, a service can be offered to monitor stock prices. Customers can express their interests in the form of queries, such as "send me alerts whenever IBM stock price is over 100". The stock prices represent the events and are continually monitored against one or more queries.

Another exemplary service can be provided to monitor public health conditions. In this case, the events are the various statistics from hospitals, doctor offices, school absentee data, and others. These data are continuously collected and monitored against one or more continual range queries. Alerts can be sent to proper agencies when one or more of such range queries match an event.

Yet another exemplary service can be provided to offer subscription services to one or more publishers. The publishers publish contents and are filtered or monitored against one or more subscriptions. The subscribers express their individual interests in the form of a continual range queries. The service providers will monitor and match the published contents

against one or more subscription queries. Matched publications are then forwarded to the subscribers.

However, it should be apparent that these examples above are only exemplary possible applications of the present invention and are not intended as limiting the present invention in any way. The present invention provides a computerized technique of monitoring events against queries and, returning back to the block diagram of Figure 1, will be of interest to any of the levels of users or implementation shown in Figure 1.

Thus, a consumer of the present invention could be considered as the end user represented as the one or more clients 111, 112 requesting the end result of the present invention, or as a service provider, represented by query monitors 121, 122, that receives a query from clients 111, 112 and provides the end result back to the clients 111, 112. Under certain conditions, it is possible that the owner/operator of the event monitors (e.g., shown as monitors 101, 102) or even the communication network 100 might be considered as the consumer of the present invention.

Exemplary Hardware Implementation

Figure 9 illustrates a typical hardware configuration of an information handling/computer system 900 in accordance with the invention and which preferably has at least one processor or central processing unit (CPU) 911.

The CPUs 911 are interconnected via a system bus 912 to a random access memory (RAM) 914, read-only memory (ROM) 916, input/output (I/O) adapter 918 (for connecting peripheral devices such as disk units 921 and tape drives 940 to the bus 912), user interface

adapter 922 (for connecting a keyboard 924, mouse 926, speaker 928, microphone 932, and/or other user interface device to the bus 912), a communication adapter 934 for connecting an information handling system to a data processing network, the Internet, an Intranet, a personal area network (PAN), etc., and a display adapter 936 for connecting the bus 912 to a display device 938 and/or printer 939 (e.g., a digital printer or the like).

In addition to the hardware/software environment described above, a different aspect of the invention includes a computer-implemented method for performing the above method. As an example, this method may be implemented in the particular environment discussed above.

Such a method may be implemented, for example, by operating a computer, as embodied by a digital data processing apparatus, to execute a sequence of machine-readable instructions. These instructions may reside in various types of signal-bearing media.

Thus, this aspect of the present invention is directed to a programmed product, comprising signal-bearing media tangibly embodying a program of machine-readable instructions executable by a digital data processor incorporating the CPU 911 and hardware above, to perform the method of the invention.

This signal-bearing media may include, for example, a RAM contained within the CPU 911, as represented by the fast-access storage for example. Alternatively, the instructions may be contained in another signal-bearing media, such as a magnetic data storage diskette 1000 (Figure 10), directly or indirectly accessible by the CPU 911.

Whether contained in the diskette 1000, the computer/CPU 911, or elsewhere, the instructions may be stored on a variety of machine-readable data storage media, such as

DASD storage (e.g., a conventional "hard drive" or a RAID array), magnetic tape, electronic read-only memory (e.g., ROM, EPROM, or EEPROM), an optical storage device (e.g. CD-ROM, WORM, DVD, digital optical tape, etc.), paper "punch" cards, or other suitable signal-bearing media including transmission media such as digital and analog and communication links and wireless. In an illustrative embodiment of the invention, the machine-readable instructions may comprise software object code.

While the invention has been described in terms of a single preferred embodiment, those skilled in the art will recognize that the invention can be practiced with modification within the spirit and scope of the appended claims.

Further, it is noted that, Applicants' intent is to encompass equivalents of all claim elements, even if amended later during prosecution.